

Wii U Planning and Operations Manual

SpotPass

2016/01/05

Version 2.0

**The content of this document is highly confidential
and should be handled accordingly.**

Confidential

These coded instructions, statements, and computer programs contain proprietary information of Nintendo and/or its licensed developers and are protected by national and international copyright laws. They may not be disclosed to third parties or copied or duplicated in any form, in whole or in part, without the prior written consent of Nintendo.

Table of Contents

1	Introduction	6
1.1	About This Document	6
1.2	Terminology	6
1.3	Related Documents	7
2	About SpotPass	8
2.1	What Is SpotPass?	8
2.2	Using SpotPass	10
2.3	What You Can Do With SpotPass	10
2.4	Applying to Use SpotPass	11
3	How BOSS Works.....	12
3.1	BOSS Tasks and the Task Database.....	12
3.2	BOSS Storage	12
3.3	Executing BOSS Tasks.....	13
3.4	Registering a BOSS Task	14
3.5	Deleting a BOSS Task	15
3.6	NBDL Tasks	15
3.6.1	What Is an NBDL Task?	15
3.6.2	Sharing of Data Instances	17
3.6.3	Deleting Files Downloaded Using NBDL.....	18
3.6.4	Avoiding Downloading the Same Data Twice.....	19
3.6.5	Other Features Provided by NBDL.....	21
3.7	RawDL Tasks	22
3.8	RawUL Tasks	24
3.9	DataStore Upload Task/Download Task	26
3.10	NBDL Data List	26
3.11	Size Required for BOSS Storage	27
4	Types of Data You Can Distribute With NBDL	28
4.1	Extra Data.....	28

4.2	Notes for Distributing Extra Data.....	28
4.3	Notifications.....	28
4.3.1	Notification Structure.....	29
4.3.2	Attaching a Mii	32
4.3.3	Displaying Notifications in WaraWara Plaza.....	32
4.3.4	Opt Out of Notifications.....	32
4.4	Notes for Distributing Notifications	33
4.4.1	Upper Limit on the Notifications List	33
4.4.2	Frequency of Notification Distribution	33
4.4.3	Maximum Number of Notifications That a Task Can Distribute at One Time.....	33
5	Restrictions on the Use of BOSS	34
5.1	Restrictions on the Amount of Data Distributed	34
5.2	Restrictions on the Intervals Between Distributions.....	34
6	User Notifications	35
6.1	Types of User Notifications	35
6.2	Default Settings for User Notifications for NBDL Tasks	35
6.3	User Notification Rules.....	35
	Revision History	37

Tables

Table 3-1	Task Execution API and Result of Each Function	13
Table 3-2	Execution Durations and Intervals of Periodic Tasks	14
Table 3-3	BOSS Task Registration Failure Cases and Solutions.....	14
Table 3-4	API Functions for Deleting Files	19
Table 3-5	API Functions to Clear Download Log When Deleting Files	21
Table 3-6	Headers Added to Requests by RawUL Tasks.....	25
Table 4-1	Items That Can Be Specified in a Notification	31
Table 5-1	Restrictions on Distribution Intervals for the Different Types of Data.....	34
Table 6-1	Types of User Notification.....	35
Table 6-2	Default Settings for User Notifications for NBDL Tasks.....	35
Table 6-3	User Notification Rules.....	35

Figures

Figure 2-1 Download Process in SpotPass	8
Figure 2-2 Planning Example: Distributing Notifications.....	9
Figure 2-3 Planning Example: Distributing Extra Data	9
Figure 2-4 SpotPass Features.....	10
Figure 2-5 SpotPass Operation	11
Figure 3-1 NBDL Task Processing.....	16
Figure 3-2 Sharing Data Instances	17
Figure 3-3 File Reference Deletion and Instance Deletion.....	18
Figure 3-4 Avoiding Downloading the Same Data Twice.....	20
Figure 3-5 Running RawDL Tasks	23
Figure 3-6 Running RawUL Tasks	25
Figure 4-1 Notifications List	29
Figure 4-2 Notifications.....	30

1 Introduction

1.1 About This Document

This document describes the SpotPass mechanism provided by the Wii U. This mechanism makes it possible to send and receive data in the game's background.

This document is intended for planners, developers, and operators considering the use of SpotPass.

The SpotPass mechanism is closely related to other Wii U network features and network accounts.

Also refer to those related documents.

1.2 Terminology

- BOSS

The name of the process and library that provides SpotPass functionality. BOSS stands for "background online services."

- Task

A unit of processing by BOSS. BOSS receives task registration from applications, and executes them.

- BOSS management tool

The tool used for tasks such as BOSS testing and recording distribution in the product environment. It can be accessed from OMAS (Online Management System). See the help manual within the tool for details on how to use it.

- BOSS storage

The location where data downloaded by BOSS is stored. One area is created for each application, at the same time that the save data area is created. Applications can access BOSS storage via an API.

- Extra data

Application data distributed by BOSS. Extra data is stored in BOSS storage.

- Notifications

Data displayed on the Notifications list. Notifications are stored in a special area of BOSS storage reserved for notifications.

- NBDL tasks

A task that distributes extra data or notifications. NBDL stands for "Nintendo batch download."

- RawDL/RawUL

A feature that accesses the specified URL via HTTP or HTTPS, and downloads and uploads data.

- DataStore

An online storage feature provided by NEX. It can be used to exchange data between users. BOSS provides access to the DataStore..

- WaraWara Plaza

The space where conversations in Miiverse are displayed. One of the features of the Wii U Menu, this space is shown on the TV screen when the Wii U Menu starts. For more information, see *Wii U Planning and Operations Manual – Miiverse*.

1.3 Related Documents

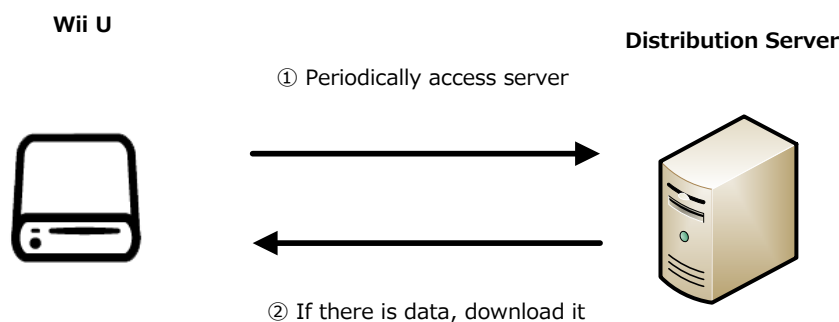
- *Wii U Account System Developer's Guide*
- *Wii U Overview: Communication Features*
- *Wii U Guidelines*

2 About SpotPass

2.1 What Is SpotPass?

SpotPass is a service that automatically uploads and downloads data in the background while an application is running. The Wii U system provides the SpotPass service. Applications can use this service via the BOSS library. When an application registers a task with BOSS, BOSS periodically accesses the server to download and upload data.

Figure 2-1 Download Process in SpotPass

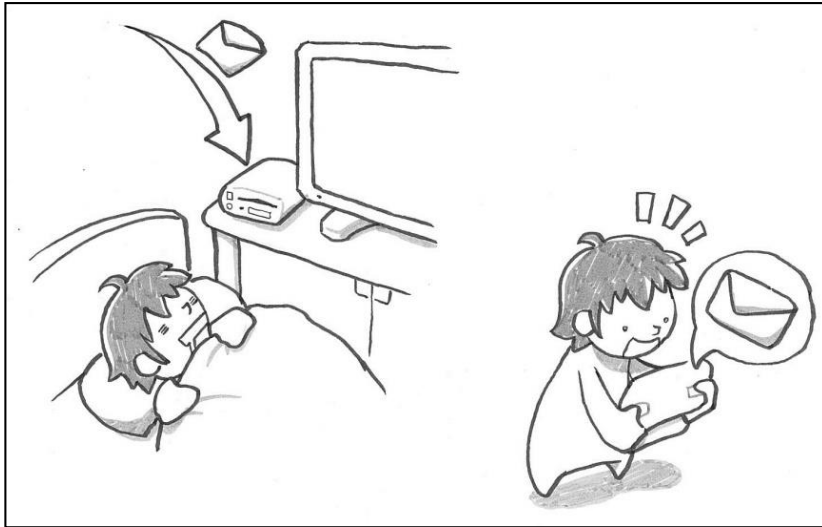


You can use this feature to provide notifications and application-specific extra data "on the spot," without forcing users to explicitly attempt getting the data. For example, say you have a game application that sends quizzes to users, which they answer. Using the SpotPass feature would let you do the following.

1. Distribute additional quizzes to all users.

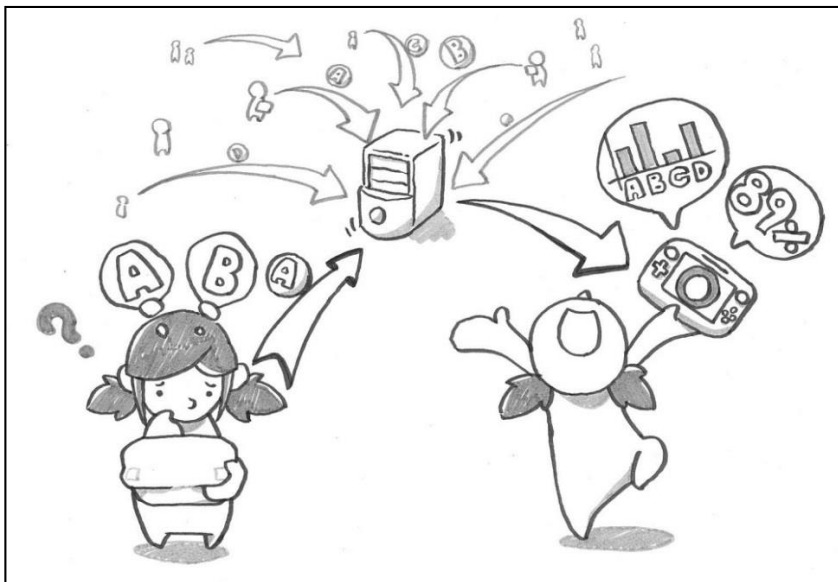
a) Distribute notifications to users about additional quizzes, including the date and time when they are available (NBDL task)

b) Distribute data for additional quizzes from Nintendo's servers (NBDL task)

Figure 2-2 Planning Example: Distributing Notifications

2. Get quiz results from independent servers.

- a) Send answers to an independent server (RawUL task)
- b) Download quiz scores and answers from the independent server (RawDL task)

Figure 2-3 Planning Example: Distributing Extra Data

Note: Data can be sent and received using SpotPass even when the Wii U is turned off.

2.2 Using SpotPass

Your application can use SpotPass by registering tasks with BOSS. A task is a unit of work. The application sends work requests to BOSS in the form of tasks, and BOSS executes these tasks.

2.3 What You Can Do With SpotPass

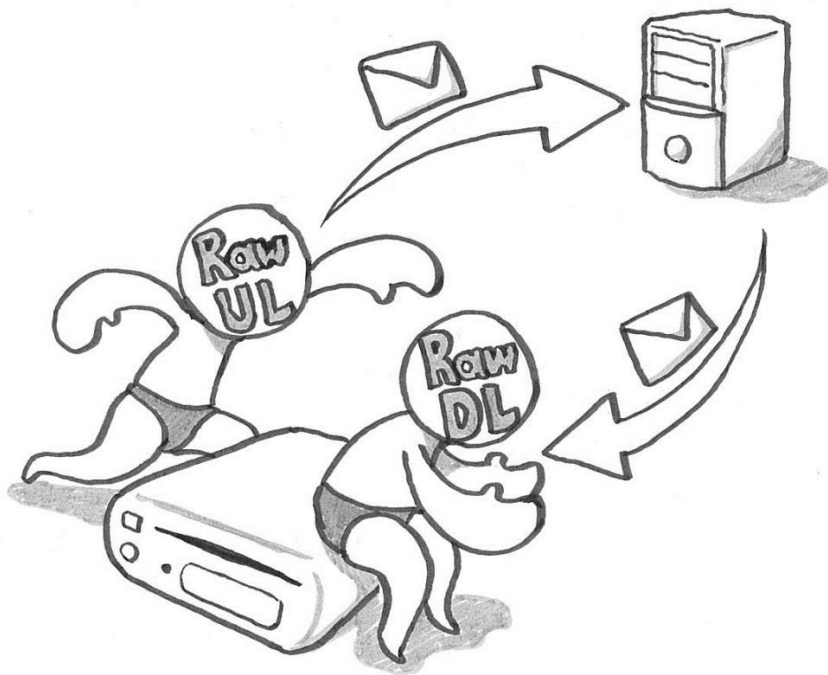
SpotPass includes the following features.

- NBDL
- RawDL
- RawUL

Figure 2-4 SpotPass Features



All of these features are managed using tasks as a unit. The following chapters describe these tasks and how BOSS works.

Figure 2-5 SpotPass Operation

2.4 Applying to Use SpotPass

To use SpotPass features, it is necessary to submit an application to Nintendo. Use OMAS to make applications. Once the application is received and approved, Nintendo will provide the required development environment.

Note: For titles sold in more than one region, you must submit an application and obtain permission to use SpotPass for each region.

3 How BOSS Works

3.1 BOSS Tasks and the Task Database

BOSS tasks are stored in the task database.

Users can create up to 12 accounts on a single Wii U console. When a user creates an account, the console creates a task database for that account in a dedicated region of NAND, the internal flash memory. The console creates a separate task database for each account.

BOSS tasks are always associated with an account. Note that it is not possible to register a single task that is shared by all accounts. An application can register up to eight tasks for a single account. Your application needs to plan a service that can be implemented with the adequate number of tasks.

The application must specify a task ID of up to seven alphanumeric characters for each task. Task IDs must be unique within your application. Nintendo manages all task IDs. You need to request your task IDs in OMAS, and at the same time Nintendo will confirm that the requested task IDs are the same as the requested task IDs during Lotcheck.

3.2 BOSS Storage

If BOSS obtains a file as a result of executing a task, it stores that file in an area called "BOSS storage." When uploading data, BOSS also keeps the data temporarily in BOSS storage while the send is queued.

BOSS storage is created at the same time the save data is created; in other words, when the `SAVEInitSaveDir` function is called. The size of the created BOSS storage is specified in the ROM header. If the specified size is too large, it fills up the storage area, and if it is too small, BOSS cannot download data. Specify an appropriate size for the data that you plan to distribute. BOSS also stores the metadata used to manage data in BOSS storage. You must also specify enough space for the data you actually intend to distribute, plus this metadata.

Internally, BOSS storage is essentially identical to save data. It has the following directory structure.

```
./account1
/account2
:
/account12
/common
```

BOSS storage has directories for each account, in addition to a `common` directory that is shared by all accounts. The ROM header must specify both the size of each account directory and the size of the `common` directory. However, BOSS does not use these directories in the same way as save data; instead, its usage of the directories depends on the types of tasks your application uses. You must specify sizes that are appropriate for the features your application will use. To learn how to specify the size, see Section 3.11 Size Required for BOSS Storage.

3.3 Executing BOSS Tasks

The application can choose whether to execute a BOSS task just once, or periodically. The functionality differs depending on which API function you call to execute the task. See the following table for details.

Table 3-1 Task Execution API and Result of Each Function

API Function	Execution Type	Result When Communication Ends
<code>Task::Run(true)</code>	Runs one time. The function runs as soon as it is called.	Returns an error.
<code>Task::Run(false)</code>	Runs one time. The function runs as soon as it is called.	Resumes execution when the network connection is restored.
<code>Task::StartScheduling(true)</code>	Runs periodically. The function runs as soon as it is called.	Resumes execution when the network connection is restored.
<code>Task::StartScheduling(false)</code>	Runs periodically. It runs for the first time after the specified interval has elapsed from the time the function was called.	Resumes execution when the network connection is restored.

Use `Task::Run(true)` if you want to wait for the task to finish; for example, if you want your application to display progress while the data downloads, and display an error if the connection is lost before the download finishes. Use the `Task::StartScheduling` function if you want to execute a task periodically.

If you schedule a task to run periodically, BOSS executes it at the duration and interval specified as follows.

Table 3-2 Execution Durations and Intervals of Periodic Tasks

Item	Value
Execution interval	8 hours
Execution lifetime	90 days

The execution lifetime starts when the task is registered. It is controlled by the system clock. If the application is not run by the time the execution lifetime elapses, the task is not executed. You must control this time period appropriately. One example is to refresh the execution lifetime every time the application launches.

The execution interval and lifetime above are the default values. Contact Nintendo if you want to change these values.

Note: Call the `Task::RestoreLifeTime` or `Task::UpdateLifeTime` function to refresh the execution lifetime.

3.4 Registering a BOSS Task

You can only register a BOSS task if the user can use network features. In other words, you cannot register BOSS tasks if the user has not yet obtained a Nintendo Network ID. If the user does not have a Nintendo Network ID, task registration returns an error. You must take appropriate action to handle this, such as displaying an error message based on the error code, or prompting the user to obtain a Nintendo Network ID. Additionally, you must handle cases where BOSS task registration fails. The following table describes the methods for handling these cases.

Table 3-3 BOSS Task Registration Failure Cases and Solutions

Case	Solution
User has not obtained a Nintendo Network ID.	Get a Nintendo Network ID.
Game Software Communication Features is restricted in the Parental Controls.	Disable this restriction in Parental Controls.
The maximum number of tasks is already in the task database.	Have the user delete a different application task from Download Management.

3.5 Deleting a BOSS Task

There are two cases when BOSS tasks are deleted.

- When the application deletes a task.
- When the user deletes a task from Download Management.

In the second case, the user deletes all BOSS tasks registered by the application. The user never only deletes a specified subset of the tasks registered by the application.

With the exception of these two cases, registered BOSS tasks are never deleted. On a Nintendo 3DS system, if the task database is full when an application tries to register a task for SpotPass, it deletes the oldest task and allows the new one to be registered. On a Wii U console, however, an error would be returned in this situation.

Note: Task information remains in the database even if their execution lifetime has elapsed before they were executed.

3.6 NBDL Tasks

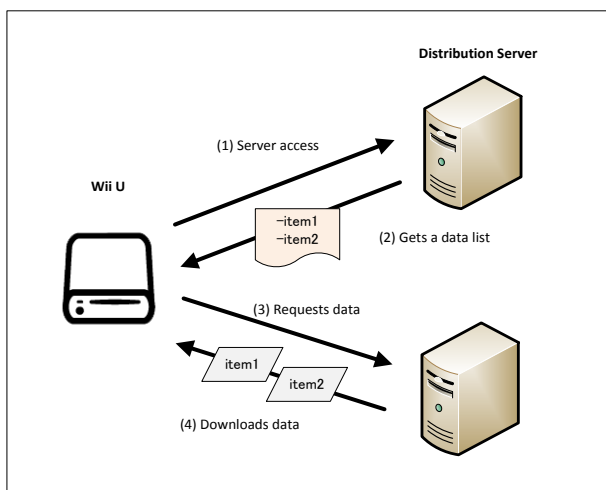
3.6.1 What Is an NBDL Task?

Nintendo batch download (NBDL) tasks are a mechanism for distributing data. This mechanism replaces the Nintendo archive download (NADL) task provided by the Nintendo 3DS system. NBDL tasks can be used to distribute extra data used by application and notifications.

Nintendo provides a complete environment for NBDL tasks, including a system for distribution. When using NBDL, Nintendo provides a web-based BOSS management tool. Application developers use the web interface to register data to distribute. The Wii U console downloads registered data from the Nintendo servers.

Below is a brief description of NBDL task processing.

1. The application registers an NBDL task.
2. When the time of execution arrives, BOSS sends the registered application ID and task ID to the server, and gets a list of files to download.
3. One at a time, BOSS downloads all the files on the list that have not been downloaded.

Figure 3-1 NBDL Task Processing

Application developers can distribute files to users simply by registering them for the target task using the BOSS data server management tool.

For example, say that you have registered two extra data files for a particular task (`task1`).

```
task1
- item1
- item2
```

In this case, BOSS would download two items: `item1` and `item2`. Now, say that you have added a new item: `item3`.

```
task1
- item1
- item2
- item3 (newly added)
```

In this case, if the user already has `item1` and `item2`, BOSS only downloads `item3`. Conversely, if the user registers the task after `item3` is added, that user will not yet have `item1` or `item2`. In this case, BOSS downloads all three registered items: `item1`, `item2`, and `item3`.

If the contents of a previously downloaded file have changed, BOSS downloads it again even if the filename is the same. As an example, let us assume that `item2` has been changed.

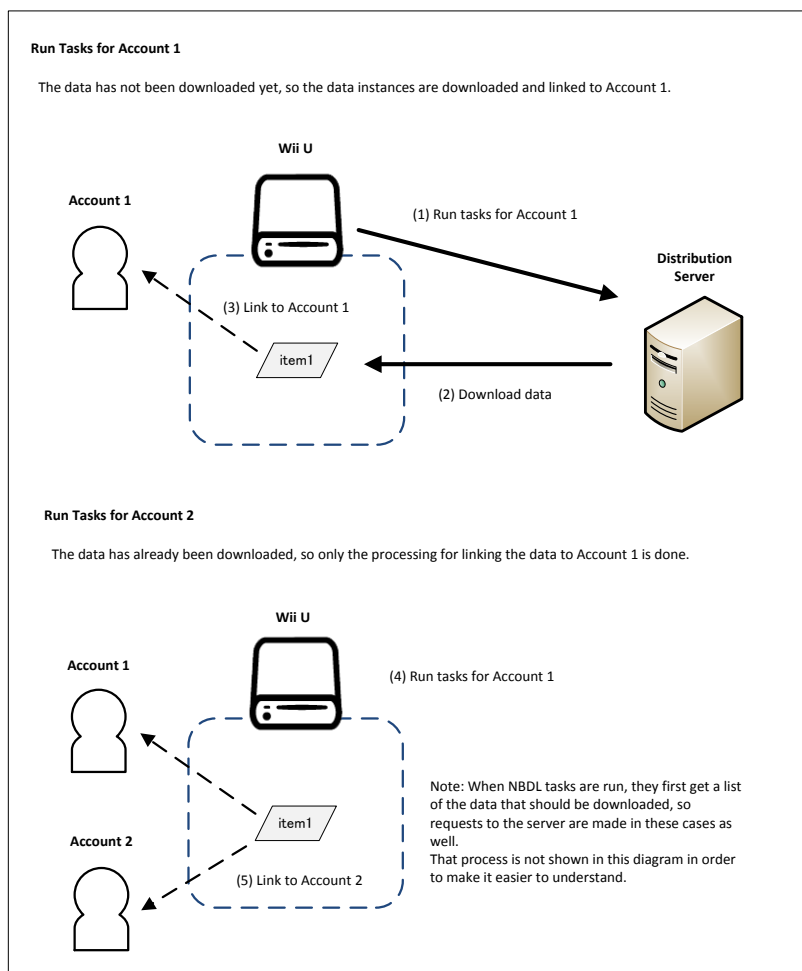
```
task1
- item1
- item2 (file was changed)
- item3
```


In this example, if the user has already downloaded all the files, the newly changed file (*item2*) is downloaded. In addition to the filename, BOSS assigns a unique ID to extra data registered using the BOSS data server management tool. It uses this ID to perform the handling described above.

3.6.2 Sharing of Data Instances

Instances of data distributed via NBDL tasks are shared between accounts. This practice conserves storage and bandwidth.

Figure 3-2 Sharing Data Instances



To accomplish this, BOSS stores all data downloaded through NBDL tasks in the `common` directory of BOSS storage. BOSS manages references for each account: bookkeeping information about which account owns which files. BOSS also stores these references in the `common` directory. If `account1` and `account2` download the same file, the file is actually only downloaded by the first account. Nothing is stored in the directories of individual accounts.

```

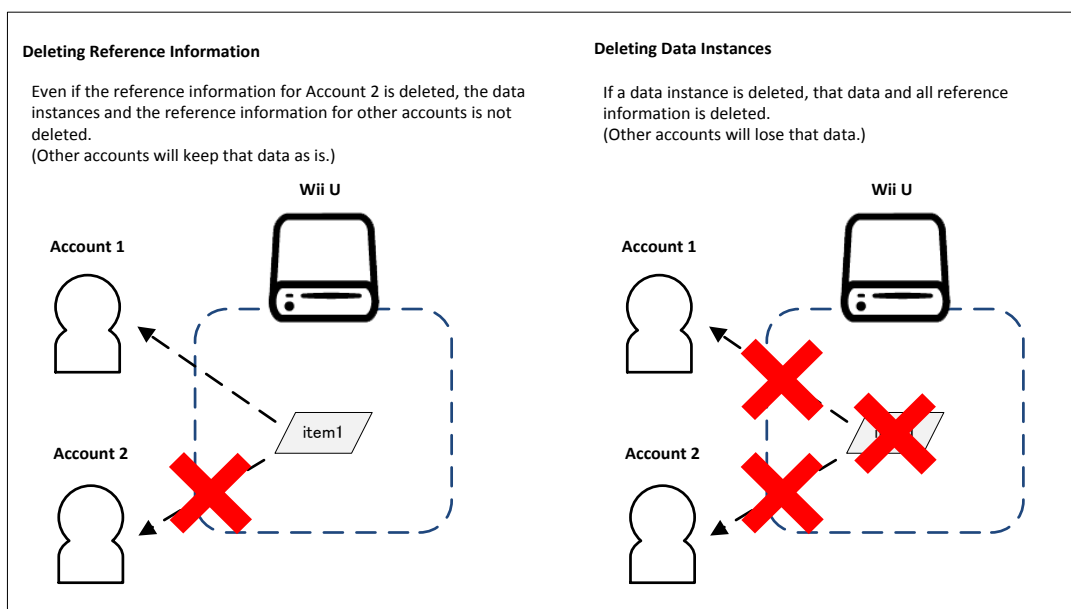
./account1/
/account2/
:
/account12
/common/account1/{file ownership information} (owns item1 and item2)
    /account2/{file ownership information} (owns item1)
    /data/item1 (holds file instance)
        /item2

```

3.6.3 Deleting Files Downloaded Using NBDL

BOSS storage is structured so that each account only keeps references to files, and the actual files are shared. The API provides two functions for deleting files: one that actually deletes the file instance, and one that only removes the reference.

Figure 3-3 File Reference Deletion and Instance Deletion



The following table shows what happens when each of these API functions is called.

Table 3-4 API Functions for Deleting Files

Function	Description
<code>NSData::Delete(void)</code>	Only deletes the reference to the file. This function does not delete the actual file instance.
<code>NSData::DeleteRealFile(void)</code>	Deletes the file instance. It also removes the ownership information of all accounts referencing this file.

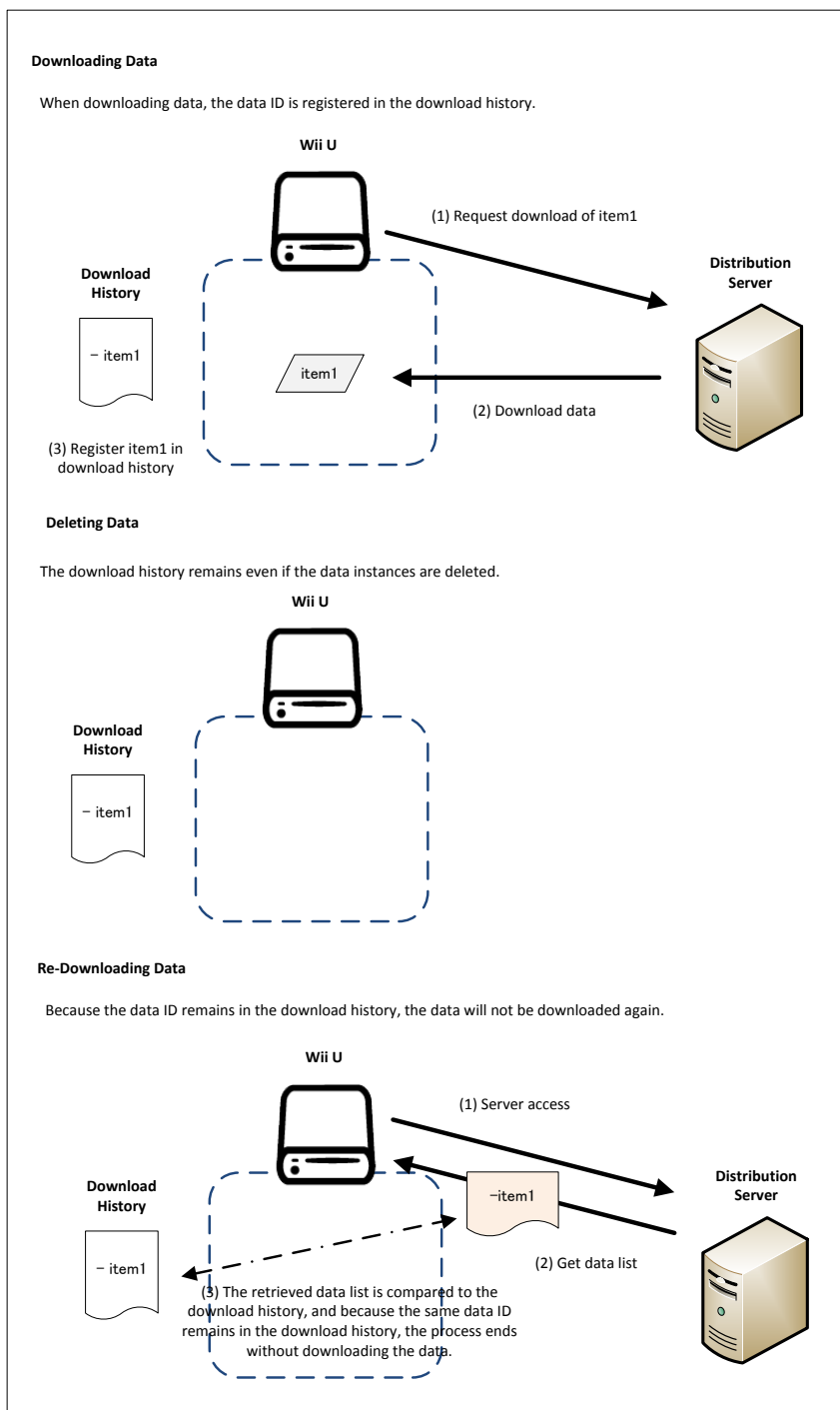
If storage is full when processing an NBDL task, BOSS deletes one or more files from storage before downloading the new file. BOSS deletes files from storage using the following rules.

- Files that are not referenced by any accounts are deleted.
- If there are no unreferenced files, files are deleted by time of download, oldest to newest.

The application can configure a downloaded file so that it is not automatically deleted. BOSS does not automatically delete files that are so configured. If storage is full, and all remaining files are configured to not automatically be deleted, BOSS returns an error without downloading the new file.

3.6.4 Avoiding Downloading the Same Data Twice

The BOSS feature assigns a unique ID to each extra data file. BOSS keeps a download log of the IDs of every file downloaded. A file's unique ID is not removed from this download log if the file is automatically deleted or deleted using one of the API functions described previously. This mechanism prevents BOSS from downloading the same data twice.

Figure 3-4 Avoiding Downloading the Same Data Twice

Note: The download log is stored in the BOSS storage created by the application.

BOSS also provides an API function to delete a file and remove its ID from the download log at the same time. If you use this function, BOSS downloads the same file again if it has been deleted and is still being distributed.

Table 3-5 API Functions to Clear Download Log When Deleting Files

Function	Description
<code>NsData::DeleteWithHistory(void)</code>	Deletes a file reference, and removes the unique ID of that file from the download log.
<code>NsData::DeleteRealFileWithHistory(void)</code>	Deletes a file instance and its references, and removes the unique ID of that file from the download log.

3.6.5 Other Features Provided by NBDL

Other features are available with NBDL beyond those described so far.

- Distribution by Country/Language

When BOSS uses an NBDL task to retrieve a file, it sends the Wii U console's country and language settings together with the request. You can use this information to distribute different files in response to requests for the same filename.

The North American and European versions of the BOSS data server management tool include an item for setting the language of the distribution region. When a language is set for the distribution region, data can only be distributed to consoles for which that language has been set as the Wii U system language.

You can use this feature to provide different notification text for each language.

For the country information, the value from the account that registered the task is sent. For the language information, the value set in the System Settings is sent.

Note: If you are going to distribute by country, you must first consult with Nintendo on the Message Board in OMAS about controlling distribution of the application, and you must get permission to distribute it in this manner.

- Specifying Period of Availability

You can set the dates that publication of a file begins and ends. If you specify a start and end date for a file, that file is distributed during the specified period.

- Test Distribution

Nintendo provides a feature to connect to a test distribution server by using a special DNS configuration.

You can use this DNS configuration to test the application on both retail and development hardware before distributing data to your users.

- Access Log

The servers maintain access logs for each title. Nintendo provides a service for downloading these logs from the BOSS management tool.

You can view this access log to determine how many times the servers were accessed to obtain extra data, and the number of unique users.

3.7 RawDL Tasks

RawDL tasks obtain data from a specified URL. The RawDL feature is designed for use when you want to download then distribute data from an independent server, rather than from a Nintendo server.

You can use either the HTTP or HTTPS protocol with RawDL. BOSS performs a GET on the specified URL, and stores the response as is in the directory of each account in BOSS storage. For example, if you obtain data with a task named `rawd1`, BOSS downloads and saves the data to a directory with the same name as the task, with the default filename ("`rawcontent.dat`").

```
./account1/rawdl/rawcontent.dat
/account2
:
/account12
/common
```

Although the application can modify the default filename; even if you change the file on the server, it will still be stored in the Wii U console with the same name: either the default filename or the name you specify. It is also not possible to distribute more than one file with a single task. If you want to distribute multiple files at the same time, you must register separate tasks for each file.

If the server returns a response of success (200 OK), and the response header contains a `Last-Modified` field, then BOSS retains that date and time, and sets it in the `If-Modified-Since` header of its next request. For this reason, Nintendo strongly recommends that your server support the following for the URLs specified by RawDL tasks.

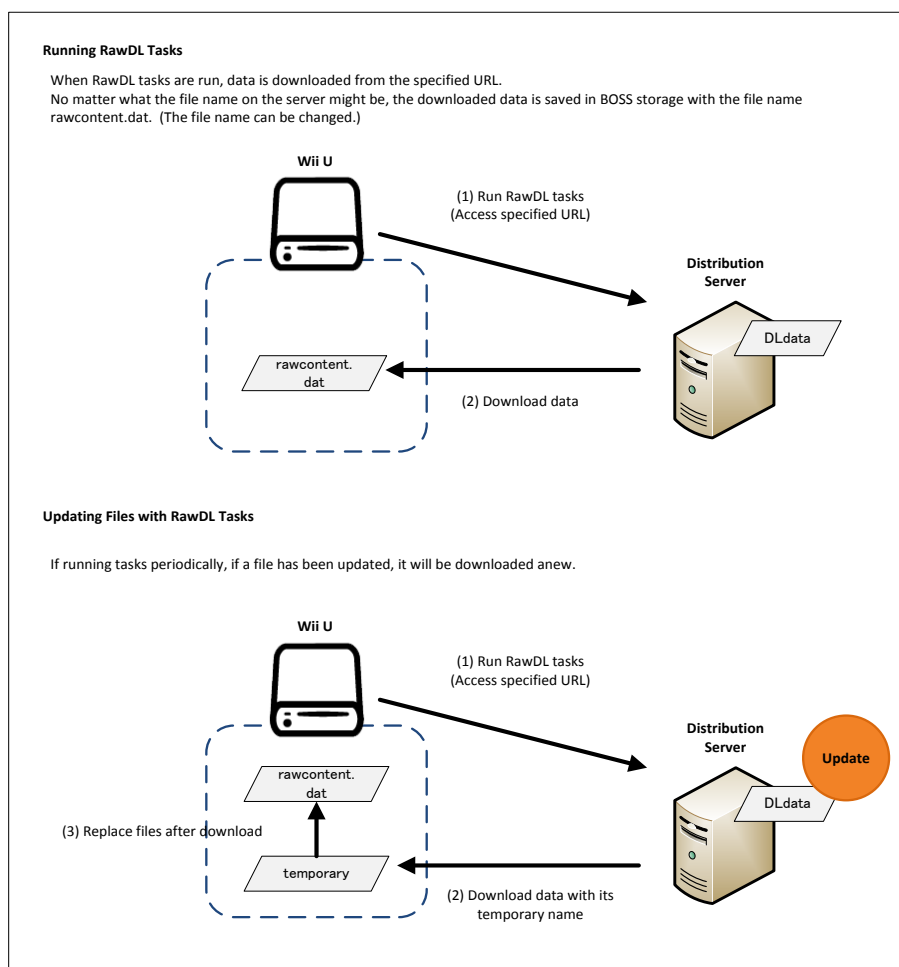
- Include a correct `Last-Modified` in the response.

- Include a correct `If-Modified-Since` in the response.

If your implementation returns a `200 OK` for every `If-Modified-Since` header from BOSS, it reduces the storage lifetime and consumes bandwidth, resulting in frequent user notifications. If the data has not been updated, respond with `304 Not Modified`.

For files that have been updated, RawDL tasks first download the data with its temporary filename, and then replace the file after the download finishes. For this reason, BOSS storage needs an amount of space equal to twice the size of the largest file, so keep track of this space requirement.

Figure 3-5 Running RawDL Tasks



RawDL tasks do not provide the following features.

- Distributing notifications
- SSL client authentication using the built-in certificate
- Preventing the same data from being downloaded multiple times, as with NBDL

Although it does not have SSL client-authentication capabilities, this task provides the following means for servers to validate clients.

- When the application registers a task, it gets a service token from the account server.
- It puts the obtained service token in the header of the HTTP request you send to the server.
- Have the server check this service token.

When you obtain a service token, the client has been authenticated, so this token can be used to validate the client. The date and time that the service token was issued are stored, so the server can determine how recently the client was validated by checking this date and time. Although the client is more secure the more recently the service token was issued, this benefit is lost if it was issued too recently, because BOSS executes periodically, even when the application is not running. Determine whether to check the date and time that the service token was issued, and if so, how old of a token you will tolerate, based on the security policy of your independent server. If you are using a service token, Nintendo recommends using an SSL connection for communication.

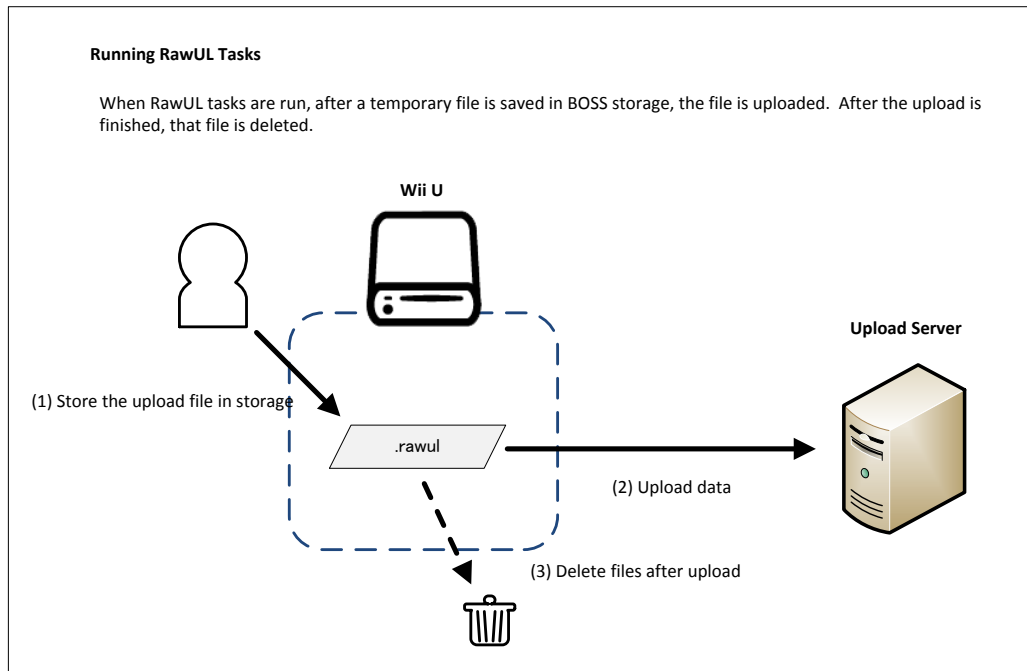
Note: Use the `NetTaskSetting::SetServiceToken` function to register a service token.

Note: To use RawDL tasks, a written HTTP Socket Library application must be submitted in addition to the application submitted in OMAS. Contact Nintendo for more information.

3.8 RawUL Tasks

A RawUL task takes a designated URL and a data file, and sends the data to the URL via HTTP POST.

BOSS temporarily stores the file to upload in the `account` directory in BOSS storage. It deletes this file when the upload is complete.

Figure 3-6 Running RawUL Tasks

For example, if you execute a task named `rawul`, BOSS creates a temporary file as follows.

```

./account1/rawul/.rawul
/account2
:
/account12
/common
  
```

If there is not enough free space in BOSS storage, it cannot create a temporary file, and the RawUL task returns an error.

BOSS adds the following headers to the HTTP POST request for the file.

Table 3-6 Headers Added to Requests by RawUL Tasks

Item	Description
X-BOSS-Digest	SHA1 digest value of entire post body (not including headers) Note: Can also be disabled.
X-Boss-UniqueId	Unique ID of application sending post

You determine whether the HTTP POST operation was successful by checking `X-BOSS-Digest`.

Note: To use RawUL tasks, a written HTTP Socket Library application must be submitted in addition to the application submitted in OMAS. Contact Nintendo for more information.

3.9 DataStore Upload Task/Download Task

DataStore Upload Task/Download Task is a feature for using the NEX DataStore features via BOSS. By using this feature you can act through BOSS to upload data to and download data from the NEX game server.

With upload tasks you can specify the recipients of the uploaded data. For the recipients, you can specify all friends or particular people. The uploaded data is sent to the specified recipients. For information about the types of recipients, see the NEX manual.

With download tasks, the data that was uploaded by upload tasks to your account is automatically received and saved to BOSS storage. This received data is separate from the data received by other BOSS tasks. Up to 1024 sets can be stored.

The Wii U DataStore Upload Task/Download Task feature uses the NEX3x versions of DataStore. Because the 3DS uses the NEX2x versions of the DataStore features, there are slight differences in what can be done using the Wii U as opposed to the 3DS. For more information, see the NEX manual.

Note: Use NEX 3.7 or later if you are going to use DataStore Upload Task/Download Task or if you will use NEX directly in the application.

3.10 NBDL Data List

NBDL Data List is a feature for getting a list of the data registered on a server. Normally with NBDL tasks, all of the data registered on the server is downloaded and saved in the local system. But by using this feature, the application can select any of the registered data to download from the server.

As an example, one possible use of this feature would be to get a list of registered challenges for a puzzle game to choose a fun one to play.

The NBDL Data List feature also can be used to filter data based on explanatory information that cannot be obtained from the downloaded data. For more information, see *Programming Manual - SpotPass*.

3.11 Size Required for BOSS Storage

BOSS stores management information in the BOSS storage allocated by the application. For this reason, you must allocate enough BOSS storage for both the size that the application will actually use and for the size that BOSS will use for management.

If necessary, use the tool that came bundled with the SDK reference manual. By simply entering the size of the data for distribution, you enable this tool to calculate the amount of BOSS storage that needs to be allocated.

Use the Wii U Application Setup Tool to specify BOSS storage size settings.

Note: The documentation and tool are located in the Communications/BOSS (SpotPass) part of the reference manual in the SDK.

4 Types of Data You Can Distribute With NBDL

4.1 Extra Data

The format of this data is up to the application. It cannot contain programs, scripts, or other executable files.

When you submit data using the BOSS data server management tool, it is converted to a proprietary Nintendo format (which includes a signature and encryption) upon approval. Nintendo then uses its CDN to distribute the data to users' Wii U consoles via HTTPS. The application can access the decrypted data.

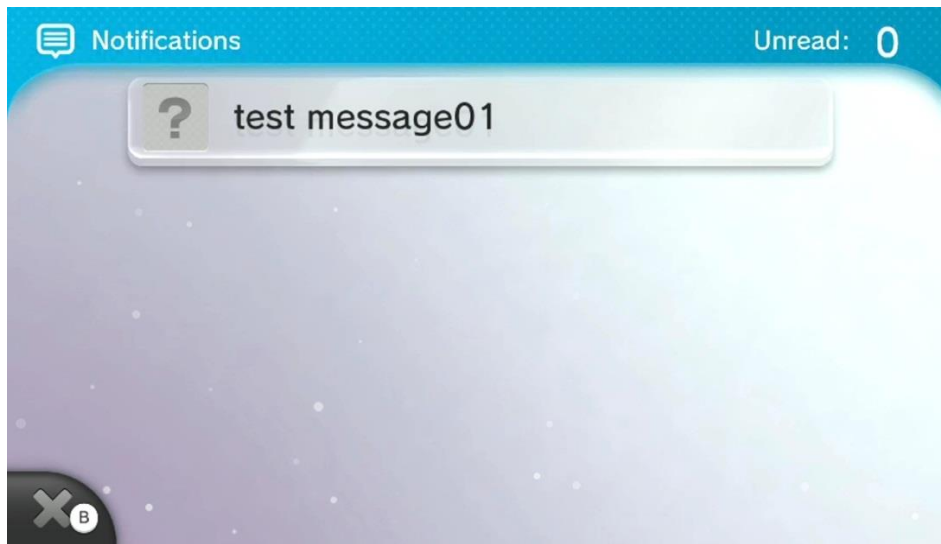
4.2 Notes for Distributing Extra Data

There are no restrictions on the size of each item of extra data that you can distribute. There are, however, restrictions on the total amount of data that can be distributed. Make sure that your distribution stays within these limits.

- The total amount of data that a single application can distribute is 512 MB per account.

4.3 Notifications

Notifications are data that appear in the Notifications list of the System Menu. As with extra data, notifications are converted into a proprietary Nintendo format upon approval, before being distributed. Figure 4-1 shows the Notifications list screen. (This screen is under development and may differ from the one provided in the actual product.)

Figure 4-1 Notifications List

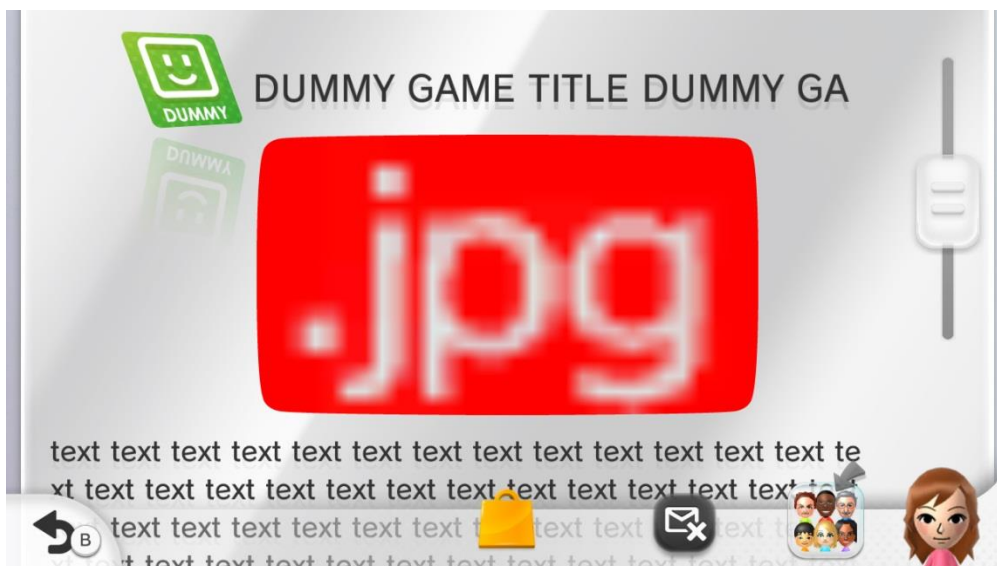
Notifications are stored in a special location on the console. Although notification tasks are registered as application tasks, the data itself is stored in BOSS storage provided by the System Menu. Although management information is also created in BOSS storage, the application does not need to create BOSS storage if it only wants to distribute the notifications.

4.3.1 Notification Structure

The basic structure of a notification consists of a subject, body, and image attachment. Specifying an image attachment is optional, but the subject and body must contain at least one character each.

Figure 4-2 is an image of the kind of notification that will actually be displayed on the Wii U. (This screen is under development and may differ from the one provided in the actual product.)

Figure 4-2 Notifications



Users can jump from a notification to the application that distributed it. You can specify parameters to pass to the application when the user jumps to it. In addition to jumping to the application, users can also jump from notifications to the web browser, Nintendo eShop, or other system applications. You can also specify jump parameters when jumping to these other destinations. For example, say that you have released add-on content for your application. If you send a notification announcing this, you could specify a jump from that notification to Nintendo eShop, and specify the page of the application's add-on content as a jump parameter to send the user directly to the sales page for that content. You can only specify one application as a jump destination from a notification, including the application that is distributing the notification. As an example, you could not register both your application and Nintendo eShop as jump destinations.

Table 4-1 Items That Can Be Specified in a Notification

Item	Description
Notification subject	1 to 31 characters (required)
Body	1 to 2999 characters (required)
Image attachment	• Image size: 640x360 • JPEG • The maximum size is 400 KB before being Base64-encoded
Parameter	• A parameter to pass when starting the distributing application • 16-digit hexadecimal number
Application launch type	Type of application that the notification launches. You can specify one of the following. • Do not launch • Distributing application • Internet Browser • Nintendo eShop • Miiverse
Parameter passed to Internet browser	Specify a URL of up to 1023 characters
Parameter passed to Nintendo eShop	The title name (see Note below)
Parameter passed to Miiverse	The title name (see Note below) and community ID

Note: Select a title name from those set on the server.

Reference: Recommended Settings for Attaching JPEG Images to Notifications

There are recommended settings for saving JPEG images attached to notifications. The following procedure uses Photoshop CS5.1 as an example:

1. From the menu, select **File > Save As**.
2. When the dialog box asks you for the file format, select **JPEG (*.JPG, *.JPEG, *.JPE)**.
3. Select the following JPEG options.
 - Image options (quality): 10, High
 - Format options: Baseline (standard)

Due to the specifications of the JPEG library used by the Notifications application, these settings provide the most consistent image quality for the Wii U GamePad and the TV.

4.3.2 Attaching a Mii

You can attach a maximum of one Mii to notifications. Only what are called "Special Mii characters" can be attached; ordinary Mii characters cannot be attached. Only Nintendo can create Special Mii characters. Contact Nintendo if you want to attach a Special Mii.

In the Notification list, this Special Mii can be saved to Mii Maker. Whether the Mii can be saved is something that can be specified when the notification is registered; it is possible to disallow saving the Mii.

When attaching a Mii, the Mii character's expression and the data for the clothing for it to wear can be set in addition to the data for the Special Mii.

4.3.3 Displaying Notifications in WaraWara Plaza

Notifications received can be displayed in the WaraWara Plaza shown in the System Menu. If you want a notification to be displayed in WaraWara Plaza, turn the display setting ON when creating the notification data.

If a Mii is attached to the notification, it appears and announces the arrival of the notification. If no Mii is attached, the default Mii announces the arrival of the notification.

Only notifications that the user has not yet read are shown in WaraWara Plaza. If there are multiple unread notifications from the same application, only the most recent is shown.

4.3.4 Opt Out of Notifications

When users receive notifications, they have the option of making a setting to no longer receive notifications from the application in question. This feature is called Opt Out. If the user elects to opt out, BOSS discards the notification data after downloading it. For this reason, the user never sees the notification.

If you want to send notifications to users after they have opted out, change the Opt Out flag setting after confirming within the application that the user approves of this change.

Note: Even if the user opts out, any notifications received before opting out continue to be shown in the Notifications list.

4.4 Notes for Distributing Notifications

4.4.1 Upper Limit on the Notifications List

The Notifications list saves up to 100 notifications. After the limit of 100 notifications is reached, notifications are deleted starting from the oldest ones. Also, if there are many notifications that come with large images attached, there may be cases where the notifications start getting deleted before the number of notifications reaches 100. Notifications come into the Notifications list from all types of applications, so the more applications are sending notifications, the higher the probability becomes that data will be deleted.

When distributing notifications, consider the content with the understanding that users quickly delete notifications. It would probably be best to avoid distributing notifications with content that users are likely to need to save.

4.4.2 Frequency of Notification Distribution

If notifications are sent frequently, customers might come to think of them as spam. That would not be favorable for the customer or for the application that would like to send notifications.

For that reason, if you are sending notifications, limit them to a distribution frequency of about one per week, on average.

4.4.3 Maximum Number of Notifications That a Task Can Distribute at One Time

The system limits the number of notifications that a task can distribute at one time to 10.

However, as noted in Section 2.3 What You Can Do With SpotPass, distributing multiple notifications at the same time is discouraged. Strive to distribute only the latest notifications.

NBDL tasks distribute all extra data and notifications currently published. For example, if you have a service that adds data to distribute each week, and a user has been retrieving the data from the start, the user would only get the new data. However, if a user has just started to retrieve the data, all data distributed up to that point is obtained. If you distribute extra data and notifications together, and you add a new notification to the ones you have been distributing, all notifications up to that point are distributed. Given the nature of notifications, only new ones have any meaning, and there is little value in delivering old news. In this case, then, delete the old notifications and only distribute the new ones.

Note: The BOSS management tool does not display an error if the number of items exceeds 10. When the task runs, the top 10 items (in sort order) currently being distributed are selected automatically for distribution. There is no guarantee that these items will be the newest 10 items.

5 Restrictions on the Use of BOSS

5.1 Restrictions on the Amount of Data Distributed

Limit the total amount of data distributed using NBDL tasks to 512 MB per account. Even when you use multiple NBDL tasks, do not exceed 512 MB for the total amount of data downloaded by all tasks.

Limit the total amount of data distributed with a combination of NBDL and RawDL tasks to 4 GB per account. Do not exceed 4 GB for the total amount of data downloaded using these different kinds of download tasks.

Note: These data volume restrictions do to apply to the distribution of notifications and their attendant storage.

Note: If a task involves the distribution of data sets A, B, and C, the total amount of data distributed to each individual account by that task is equal to the sum of A, B and C. When calculating the amount of data, you do not need to consider the possibility that data might be re-downloaded.

5.2 Restrictions on the Intervals Between Distributions

If BOSS is used to distribute notifications and extra data excessively, there is a risk that users will lose interest in distributed content and the value of the distribution process itself will be lessened. For this reason, comply with the following restrictions for the intervals at which notifications and extra data are distributed.

Table 5-1 Restrictions on Distribution Intervals for the Different Types of Data

Type of Data Distributed	Restriction on Distribution Interval
Notifications	The Notifications list can store up to 100 notifications. To prevent this list from filling up with certain notifications, limit the distribution of notifications to once per week for all of the tasks registered in the application.
Extra Data	There are no particular restrictions on the distribution of extra data. However, indiscriminate use presents risks similar to that for notifications, so please settle on some distribution interval so as not to deliver extra data too frequently.

6 User Notifications

6.1 Types of User Notifications

Table 6-1 lists the means by which users are notified of the arrival of data in BOSS.

Table 6-1 Types of User Notification

Means of Notification	Description
Account icon	On the User Selection screen, the Mii icons for the accounts that have received data will be shown highlighted.
Application icon	In the Wii U Menu, the icon for the application that has received data will be shown highlighted.

For the extra data distributed with NBDL tasks, which notification type to use can be specified on the data side. An API for selecting the notification type will be provided for RawDL tasks.

6.2 Default Settings for User Notifications for NBDL Tasks

NBDL tasks can distribute notification data and extra data for use by applications. The settings for user notifications when these types of data are distributed are as follows.

Table 6-2 Default Settings for User Notifications for NBDL Tasks

Data Type	Account Icon	Application Icon
Notifications	Displayed highlighted	Displayed highlighted
Extra Data	Not displayed highlighted	Displayed highlighted

The default settings for user notifications for NBDL tasks can also be changed from the default settings listed above. Contact Nintendo if you wish to make changes.

6.3 User Notification Rules

You must follow the rules below regarding user notifications.

Table 6-3 User Notification Rules

Condition	Rule
If the account icon is highlighted	The application must be highlighted.
If the application icon is highlighted	None. (Highlighting the account icon is optional.)

For example, if you highlight the account icon, but do not want to highlight the application icon, understand that these controls are prohibited for operational reasons.

However, if you just want to distribute data without using any user notifications—that is, without highlighting either the account icon or the application icon—that is permitted.

Revision History

Version	Revision Date	Category	Description
2.0	2016/01/05	Changed	• General Made symbols uniform. Standardized the name of the tool to “BOSS management tool.” • 1.2 Terminology Added the term BOSS management tool. • 3.9 DataStore Upload Task/Download Task Added a note.
		Deleted	• 7 Using the BOSS Data Server Management Tool Deleted because the content was moved to the BOSS management tool help file.
1.4	2014/06/23	Added	• 3.11 DataStore Upload Task/Download Task • 3.12 NBDL Data List
		Changed	• General Deleted the sentence stating that DataStore is not currently supported.
1.3	2014/03/13	Added	• 5 Restrictions on the Use of BOSS

Version	Revision Date	Category	Description
		Changed	• 1.1 About This Document Deleted the note about highlighting undetermined terms in turquoise because no terms are currently highlighted. • 1.2 Terminology Added the term WaraWara Plaza. • 2.1 What Is SpotPass? Changed the Note about the use of SpotPass when the power is off. Changed the graphic in Figure 2 1. • 4.3.3 Displaying Notifications in WaraWara Plaza Deleted the Note stating that WaraWara Plaza will be supported in the future. • 6.1 Types of User Notifications Deleted references to illuminating the HOME Button LED as a way to notify users because this feature is no longer planned. • 6.2 Default Settings for User Notifications for NBDL Tasks Changed Table 6 2, deleting the column for lighting the HOME Button LED because this feature is no longer planned as a way to notify users. • 6.3 User Notification Rules Deleted references to illuminating the HOME Button LED as a way to notify users, because this feature is no longer planned.
1.2	2013/01/10	Changed	• 1.3 Related Documents The title of <i>Wii U Account Specifications</i> was changed to <i>Wii U Account System Developer's Guide</i>. • 2.4 Applying to Use SpotPass Added a note about titles being distributed in more than one region. • 3.6.5 Other Features Provided by NBDL Added information about the language set for the distribution region. Added a note about distribution by country. • 5.1 Types of User Notifications Added a note about illuminating the HOME Button LED to make user notifications. • 6.2 Flow From Data Registration to Distribution Added a description and figure for the test distribution access log. Changed the figure used for Figure 6-6. • 6.3 Registering Notifications Changed the figure used for Figure 6-22. Changed the text to match changes to figures.

Version	Revision Date	Category	Description
1.1	2012/10/15	Added	4.4.3 Maximum Number of Notifications That a Task Can Distribute at One Time
		Changed	1.1 About This Document Added a note. 3.4 Registering a BOSS Task Revised text. 4.3.1 Notification Structure Added recommended settings for image attachments. 4.3.3 Displaying Notifications in WaraWara Plaza Added a note. 4.3.4 Opt Out of Notifications Revised text. 6.4 Access Controls Replaced Figure 6-26 because of error in content.
1.0	2012/08/29	Added	- Added multiple figures and images. - Added notification images. - Added Chapter 6 Using the BOSS Data Server Management Tool.
		Changed	- Changed the name of the API function in Section 3.3 Executing BOSS Tasks. - Added the content for Chapter 5 User Notifications.
		Deleted	Deleted the section about other cautions.
0.9	2012/-6/25	—	Initial version.

All company and product names in this document are the trademarks or registered trademarks of their respective companies.

© 2012–2016 Nintendo

The contents of this document cannot be duplicated, copied, reprinted, transferred, distributed, or loaned in whole or in part without the prior approval of Nintendo.